REMARKS

Claims 1-34 are pending in the application. Claims 1-5, 7, 10, 15, 19-21, 23, 26 and 31 are rejected under 35 U.S.C. §103(a). Claims 6, 8, 9, 11-14, 16-18, 22, 24, 25, 27-30 and 32-34 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Applicants respectfully traverse the rejections for at least the reasons stated below and respectfully request that the Examiner reconsider and withdraw all outstanding rejections.

I. REJECTIONS UNDER 35 U.S.C. § 103(a):

The Office Action has rejected claims 1-5, 7, 10, 15, 19-21, 23, 26 and 31 under 35 U.S.C. § 103(a) as being unpatentable over the Admitted Prior Art (APA) in view of Russell (U.S. Patent No. 6,349,388) and further in view of Dorn et al. (U.S. Patent No. 6,012,081) (hereinafter "Dorn"). Applicants respectfully traverse these rejections for at least the reasons stated below and respectfully request the Examiner to reconsider and withdraw these rejections.

A. The Examiner has not provided any motivation for combining APA, Rusell and Dorn.

A *prima facie* showing of obviousness requires the Examiner to establish, *inter alia*, that the prior art references teach or suggest, either alone or in combination, all of the limitations of the claimed invention, and the Examiner must provide a motivation or suggestion to combine or modify the prior art reference to make the claimed inventions. M.P.E.P. § 2142. The motivation or suggestion to combine references must come from one of three possible sources: the nature of the problem to be solved, the teaching of the prior art and the knowledge of persons of ordinary skill in the art. *In re Rouffet*, 47 U.S.P.Q.2d. 1453, 1458 (Fed. Cir. 1998). The showings must be clear and particular. *In re Lee*, 277 F.3d 1338, 1343, 61 U.S.P.Q.2d 1430, 1433-34 (Fed. Cir. 2002); *In re Kotzab*, 217 F.3d 1365, 1370, 55

U.S.P.Q.2d 1313; 1317 (Fed. Cir. 2000); *In re Dembiczak*, 50 U.S.P.Q.2d 1614, 1617 (Fed. Cir. 1999). Broad conclusory statements regarding the teaching of multiple references, standing alone, are not evidence. *Id.*

In order to reject under 35 U.S.C. § 103, therefore, the Examiner must provide a proper motivation for combining or modifying the references. *In re Rouffet*, 47 U.S.P.Q.2d 1453, 1457-1458 (Fed. Cir. 1998); M.P.E.P. § 2142. The Examiner does not provide any motivation for combining references APA, Russell and Dorn. Instead, the Examiner states that the motivation for modifying APA with the teaching of Russell is to "provide a scalable approach to supporting an arbitrarily number of timers and reduces the typical processor overhead and hardware overhead involved in managing times (col. 2, lines 27-31)." Paper No. 5, page 3. This motivation is not a motivation for combining references APA, Russell and Dorn. Furthermore, the motivation stated by the Examiner is the principle of operation of the present invention described in Russell. This is not a motivation or suggestion for modifying reference APA with Russell or for combining references APA, Russell and Dorn. Further, the Examiner states that the motivation for modifying APA with the teaching of Dorn is to provide a method to modify the behavior of an application. Again, this motivation is not a motivation for combining references APA, Russell and Dorn.

There is no motivation to combine APA, Russell and Dorn. The Examiner has not established a *prima facie* case of obviousness by not providing a suggestion or motivation in either APA, Russell or Dorn, or in their combination, or in the knowledge of those ordinarily skilled in the art, to combine the teaching of a timer management program for managing timers, as taught in APA, with the teaching of a timer processing engine for supporting multiple virtual minimum time timers, as taught in Russell, with the teaching of threading control, as taught in Dorn. Russell states:

> Briefly, the present invention provides a *timer processing engine for supporting multiple virtual minimum time timers*. The plurality of

> virtual minimum time timers of the timer processing engine includes a timer data structure suitable to store the timer states of a plurality of virtual minimum time timers. Each timer state may include a last time value indicating the last time a timer was processed, an elapsed time value indicating the length of time a timer has been running, a terminal time value indicating a time at which a timer is to expire, and a set of attributes for indicating how a timer is to be processed. The timer states could be arranged in a variety of ways such as, for example, a linked list from a timer state with the shortest terminal time value to a timer state with the longest terminal time value. A timer state machine of the timer processing engine processes the timer states. Column 1, lines 50-65.
>
> In accordance with the present invention, *the timer processing engine provides a scalable approach to supporting an arbitrarily large number of timers and reduces the typical processor overhead and hardware overhead involved in managing timers.* Column 2, lines 27-32.

Thus, Russell teaches a timer processing engine that *supports multiple virtual minimum time timers*. Dorn states:

> The present invention is directed to multi-programming systems on objected oriented platforms. More particularly, *the invention is directed to threading control.* Column 1, lines 5-7.
>
> The present invention provides a *service and event synchronous/asynchronous manager (SESAM) which provides a programmer interface to concurrency, dispatching and synchronization in an object oriented computing system.* The programmer thus is relieved from bothering with low level thread details. The interface serves as a gate to synchronous/asynchronous functions. Column 2, lines 50-57.

Thus, Dorn teaches *threading control*. In particular, Dorn teaches a service and event synchronous/asynchronous manager which provides a *programmer interface to concurrency, dispatching and synchronization in an object oriented computing system*. The Examiner has not shown why the teaching of a timer management program for managing timers, as taught in APA, should be combined with the teaching of a timer processing engine for supporting multiple virtual minimum time

timers, as taught in Russell, and the teaching of a service and event synchronous/asynchronous manager which provides a programmer interface to concurrency, dispatching and synchronization in an object oriented computing system, as taught in Dorn, from either the nature of the problem to be solved, the teaching in the prior art or the knowledge of persons of ordinary skill in the art. *In re Rouffet*, 47 U.S.P.Q.2d 1453, 1458 (Fed. Cir. 1998). The Examiner must submit **objective evidence** and not rely on his own subjective opinion for combining APA, which teaches a timer management program for managing timers, with Russell, which teaches a timer processing engine for supporting multiple virtual minimum time timers, with Dorn, which teaches a service and event synchronous/asynchronous manager which provides a programmer interface to concurrency, dispatching and synchronization in an object oriented computing system. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002).

Moreover, the Examiner has not shown why APA should be modified to have a timer database for storing timer-related information from either the nature of the problem to be solved, the teaching of the prior art or the knowledge of persons of ordinary skill in the art. *In re Rouffet*, 47 U.S.P.Q.2d 1453, 1458 (Fed. Cir. 1998). Further, the Examiner has not shown why APA should be modified to have a timer services detecting the expiring of a timer where a handle function of the timer services allows the application to act on an expired timer without incurring an illegal time-out message from either the nature of the problem to be solved, the teaching of the prior art or the knowledge of persons of ordinary skill in the art. *Id.* Further, the Examiner has not shown why APA should be modified to provide a scalable approach which supports an arbitrarily number of timers and reduces the typical processor overhead and hardware overhead involved in managing timers from either the nature of the problem to be solved, the teaching of the prior art or the knowledge of persons of ordinary skill in the art. *Id.* Further, the Examiner has not shown why APA should be modified to modify the behavior of an application from either the nature of the

problem to be solved, the teaching of the prior art or the knowledge of persons of ordinary skill in the art. *Id.*

The Examiner must submit **objective evidence** and not rely on his own subjective opinion in support of modifying APA to have a timer database for storing timer-related information. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). Further, the Examiner must submit **objective evidence** and not rely on his own subjective opinion in support of modifying APA to have a timer services detecting the expiring of a timer where a handle function of the timer services allows the application to act on an expired timer without incurring an illegal time-out message. *Id.* Further, the Examiner must submit **objective evidence** and not rely on his own subjective opinion in support of modifying APA to provide a scalable approach to support an arbitrarily number of timers and reduces the typical processor overhead and hardware overhead involved in managing timers. *Id.* Further, the Examiner must submit **objective evidence** and not rely on his own subjective opinion in support of modifying APA to modify the behavior of an application. Therefore, the Examiner has not presented a *prima facie* case of obviousness for rejecting claims 1-5, 7, 10, 15, 19-21, 23, 26 and 31.

B. APA, Russell and Dorn, taken singly or in combination, do not teach or suggest the following limitations.

APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "an application program interface (API) providing a set of synchronous functions allowing an application to functionally operate a timer" as recited in claim 1. The Examiner directs Applicants' attention to the "timer management program" on page 2 of Applicants' specification as teaching the above-cited claim limitation. Paper No. 5, page 2. Instead, the timer management program, as discussed on page 2 of Applicants' specification, is a program for managing a plurality of timers. The timer management program is not an application program interface (API). Neither is the timer management program an API that provides a set of synchronous functions

allowing an application to functionally operate a timer. Accordingly, the Examiner has not presented a prima facie case of obviousness for rejecting claim 1. M.P.E.P. §2143.

APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "a timer services detecting the expiring of said timer, wherein a handle function of said timer services allows said application to act on an expired timer without incurring an illegal time-out message" as recited in claim 1. The Examiner directs Applicants' attention to column 13, line 55 to column 15, line 12 of Dorn as teaching the above-cited claim limitation. Paper No. 5, page 3. Instead, Dorn states:

> Functions that are submitted to SESAM for invocation may depend on special thread specific data settings. But *because SESAM provides an execution agent (for threads) for internal function invocation, hooks are provided allowing the application programmer to set and save thread specific data before and after function invocation.* Hooks can be specified by the application programmer upon invocation of selected SESAM API functions using a HookInfo structure. Column 8, lines 31-40.
>
> *An "addTimer()" interface allows the specification of hooks to be invoked before the timer callback is activated (beforeCb component) and after the timer callback has returned (afterCb component).* These hook functions are invoked in the same thread in which the timer callback is invoked. Column 15, lines 6-11.

Thus, Dorn teaches a synchronous/asynchronous manager providing hooks to allow the application program to set and save thread specific data before and after a function invocation. This language does not teach a *timer service detecting the expiring of a timer.* Further, this language does not teach *a handle function of the timer services.* Further, this language does not teach a handle function of the timer services that allows *the application to act on an expired timer without incurring an illegal time-out message.* Accordingly, the Examiner has not presented a *prima facie* case of obviousness for rejecting claim 1. M.P.E.P. §2143.

APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "*creating a timer from an allocated block of system memory by an application program interface (API)*" as recited in claim 20. The Examiner directs Applicants' attention to column 8, lines 41-64 of Dorn as teaching the above-cited claim limitation. Paper No. 5, pages 4-5. Instead, Dorn states:

> As illustrated, SESAM contains 5 different types of slots, each type tailored to its specific purpose:
>
> Dynamic slots 200 handle asynchrony for the application programmer,
>
> Timer slots 202 (synchronous), handle synchronous timers,
>
> Timer slots 204 (asynchronous), handle asynchronous timers,
>
> Exception slots 206 handle user defined system exception callbacks and
>
> Generic Slots, offering the comfort of waitFor . . . () functionality for user events, i.e., external event notification. Column 8, lines 51-64.
>
> Timers are used to start an activity after a specified time delay. Such an activity is most often the invocation of a user specified function. The user function is scheduled to run after the timer has expired. *Timer slots 202 and 204 (FIG. 5) are used within SESAM to support timer controlled activities.* Column 13, lines 55-60.

Thus, Dorn teaches timer slots to handle synchronous and asynchronous timers. However, this language does not teach *creating a timer*. Instead, Dorn teaches slots to support timer controlled activities. Timer slots are not timers. Furthermore, Dorn does not teach creating a timer *from an allocated block of system memory by an application program interface (API)*. Accordingly, the Examiner has not presented a *prima facie* case of obviousness for rejecting claim 20. M.P.E.P. §2143.

APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "sending a time-out message to a particular queue when said timer expires, wherein said timer is in an expired state in an asynchronous state machine, wherein a

handle function allows said application to act on said expired timer without incurring an illegal time-out message" as recited in claim 20. Instead, the Examiner directs Applicants' attention to column 13, line 55 to column 15, line 12 of Dorn as teaching the above-cited claim limitation. Paper No. 5, page 3. As stated above, Dorn teaches a synchronous/asynchronous manager providing hooks to allow the application program to set and save thread specific data before and after a function invocation. This language does not teach sending a time-out message to a particular queue when a timer expires. Further, this language does not teach a timer in an expired state in an asynchronous state machine. Further, this language does not teach a handle function that allows an application to act on a expired timer without incurring an illegal time-out message. Accordingly, the Examiner has not presented a prima facie case of obviousness for rejecting claim 20. M.P.E.P. §2143.

For at least the above reasons, claims 1 and 20 are patentable over APA in view of Russell in further view of Dorn. Claims 2-5, 7, 10, 15, 19, 21, 23, 26 and 31 each recite combinations of features including the above combinations, and thus are patentable for at least the above reasons as well. Claims 2-5, 7, 10, 15, 19, 21, 23, 26 and 31 recite additional features which, in combination with the features of the claims upon which they depend, are patentable over APA in view of Russell in further view of Dorn.

For example, APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said application performs the following operations on said timer via said API: creating said timer from an allocated block of said memory; activating said timer" as recited in claims 2-4. The Examiner directs Applicants' attention to column 8, lines 41-64, column 13, line 55 – column 15, line 12 as teaching the above-identified limitations. Paper No. 5, pages 3-4. Instead, as stated above, Dorn teaches timer slots to handle synchronous and asynchronous timers. However, this language does not teach creating a timer. Instead, Dorn teaches slots to support timer controlled activities. Timer slots are not timers. Furthermore, Dorn

does not teach creating a timer from an allocated block of memory. Furthermore, Dorn does not teach creating a timer from an allocated block of memory by an application via an API. Furthermore, this language does not teach activating a timer by an application via an API. Accordingly, the Examiner has not presented a prima facie case of obviousness for rejecting claims 2-4. M.P.E.P. §2143.

APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "*reinitiating said timer using said allocated block of system memory*" as recited in claim 2. The Examiner states that "Dorn does not explicitly teach reinitializing the timer using the allocated block of system memory. Dorn teaches the slot is reusable. It would have been obvious to apply the teaching of Dorn to the system of APA because it provides the programmers not to bother with the low level details." Paper No. 5, page 4. As stated above, the timer slot taught by Dorn does not correspond to a timer. Furthermore, the Examiner admits that Dorn and the other references, APA and Russell, do not teach the above-identified limitation. Yet the Examiner asserts that by simply combining APA, Russell and Dorn, the limitation is taught. Applicants respectfully assert that the Examiner must first submit evidence that the above-identified limitation is taught or suggested in any one of the above-identified references or is well known in the art. M.P.E.P. §2143-2144. Further, the Examiner must submit **objective evidence** and not rely on his own subjective opinion in support of combining APA, Russell and Dorn. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). Furthermore, Applicants respectfully traverse any suggestion that reinitiating a timer using an allocated block of system memory is a low level detail. Further, Applicants respectfully direct the Examiner's attention to column 2, lines 54-55 of Dorn which states that one of the benefits of the invention in Dorn is that "the programmer...is relieved from bothering with low level ***thread*** details." Reinitiating a timer using an allocated block of system memory is not a low level thread detail. Accordingly, the Examiner has not presented a prima facie case of obviousness for rejecting claim 2. M.P.E.P. §2143.

APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said timer expires and said *timer services sends synchronously a time-out message to said application*, wherein said *time-out message is sent using said allocated block of system memory*" as recited in claim 3. The Examiner states that "Dorn does not explicitly teach the time-out message is sent using the allocated block of system memory. Dorn teaches the time-out message is sent to an internal queue (the expiration event...internally; col. 14, lines 35-39). It would have been obvious to apply the teaching of Dorn to the system of APA because it provides the programmers not to bother with the low level details." Paper No. 5, page 4. Instead, Dorn states:

> FIG. 10 illustrates a sequence of activities related to interval timers. As illustrated, if an *interval timer expires before the according callback is invoked, the expiration events will be queued internally.* Column 14, lines 35-39.

Thus, Dorn teaches that if an interval timer expires before the invocation of the callback function, the expiration events will be queued internally. This language does not teach a *time-out message*. Neither does this language teach a time-out message *sent to an application synchronously* by *timer services*. Further, this language does not teach that a *time-out message is sent using an allocated block of system memory*. Furthermore, the Examiner admits that Dorn and the other references, APA and Russell, do not teach that the time-out message is sent using an allocated block of system memory. Yet the Examiner asserts that by simply combining APA, Russell and Dorn, the limitation is taught. Applicants respectfully assert that the Examiner must first submit evidence that this limitation is taught or suggested in any one of the above-identified references or is well known in the art. M.P.E.P. §2143-2144. Further, the Examiner must submit **objective evidence** and not rely on his own subjective opinion in support of combining APA, Russell and Dorn. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). Furthermore, Applicants respectfully traverse the assertion that sending a time-out message using an allocated block of

system memory is a low level detail. Further, Applicants respectfully direct the Examiner's attention to column 2, lines 54-55 of Dorn which states that one of the benefits of the invention in Dorn is that "the programmer...is relieved from bothering with low level ***thread*** details." Sending a time-out message using an allocated block of system memory is not a low level thread detail. Accordingly, the Examiner has not presented a prima facie case of obviousness for rejecting claim 3. M.P.E.P. §2143.

APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said timer expires and said timer services sends a time-out message to a particular queue, wherein said timer is in an expired state in an asynchronous state machine" as recited in claim 4. The Examiner directs Applicants' attention to page 2, lines 1-19 of APA as teaching the above-identified limitation. Paper No. 5, page 4. Instead, APA teaches that the user of the timer may be notified when the timer expires. However, this language does not teach timer services sending a time-out message to a particular queue. Further, this language does not teach that a timer is in an expired state in an asynchronous state machine. Accordingly, the Examiner has not presented a prima facie case of obviousness for rejecting claim 4. M.P.E.P. §2143.

APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said *particular queue is a system queue attached to said application*" as recited in claims 5 and 21. The Examiner directs Applicants' attention to column 14, lines 35-39 of Dorn as teaching the above-identified limitation. Paper No. 5, page 5. Instead, as stated above, Dorn teaches that if an interval timer expires before the invocation of the callback function, the expiration events will be queued internally. This language does not teach a *system queue attached to an application.* Further, the Examiner states that "it would have been obvious to modify the queue in the system of Dorn to attach to the application in the system of APA *because it serves the same purpose.*" Paper No. 5, page 5. This is not a proper motivation for modifying Dorn. The Examiner must submit **objective evidence** and not rely on his

own subjective opinion in support of combining APA, Russell and Dorn. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). Accordingly, the Examiner has not presented a prima facie case of obviousness for rejecting claims 5 and 21. M.P.E.P. §2143.

APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said application stops said timer, wherein said timer is in an idle state in said asynchronous state machine with said time-out message being queued" as recited in claim 7 and similarly in claim 23. The Examiner directs Applicants' attention to column 2, lines 10-19 of Dorn as teaching the above-identified limitation. Paper No. 5, page 5. Instead, Dorn states:

> Thread--a parallel execution unit within a process. A monitor synchronizes, by forced sequentialization, the parallel access of several simultaneously running threads, which all call up functions of one object that are protected through a monitor.
>
> Synchronizations-Primitive--a means of the operating system for reciprocal justification of parallel activities.
>
> Semaphore--a Synchronizations-Primitive for parallel activities.
>
> Mutex--a special Synchronizations-Primitive for parallel activities, for mutual exclusion purposes, it includes a critical code range. Column 2, lines 7-19.

Thus, Dorn teaches a list of terms including a thread, synchronizations, semaphore and mutex. None of these terms teaches an *application stopping a timer*. Further, none of these terms teaches a *timer in an idle state in an asynchronous state machine*. Further, none of these terms teaches a timer in an idle state in an asynchronous state machine *with a time-out message being queued*. Accordingly, the Examiner has not presented a prima facie case of obviousness for rejecting claims 7 and 23. M.P.E.P. §2143.

APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said timer is activated by said application, wherein said timer is in a

running state in said asynchronous state machine with said time-out message being queued" as recited in claim 10 and similarly in claims 15, 26 and 31. The Examiner states that "APA does not explicitly teach the timer is activated by the application, wherein the timer is in a running state in the asynchronous state machine with the time-out message being queued. However, APA teaches the application stop the timer while the time-out message being queued. It would have been obvious to one of ordinary skill in the art, the application could continue send the activate request to the timer while the time-out message being queue." Paper No. 5, page 5. Applicants respectfully traverse the assertion that APA teaches the application stops the timer while the time-out message being queued. Further, the Examiner admits that the references, APA, Russell and Dorn, do not teach that the timer is activated by the application where the timer is in a running state in an asynchronous state machine with the time-out message being queued. Yet the Examiner asserts that by simply combining APA, Russell and Dorn, the limitation is taught. Applicants respectfully assert that the Examiner must first submit evidence that this limitation is taught or suggested in any one of the above-identified references or is well known in the art. M.P.E.P. §2143-2144. Further, the Examiner must submit **objective evidence** and not rely on his own subjective opinion in support of combining APA, Russell and Dorn. In re Lee, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). Accordingly, the Examiner has not presented a prima facie case of obviousness for rejecting claims 10, 15, 26 and 31. M.P.E.P. §2143.

APA, Russell and Dorn, taken singly or in combination, do not teach or suggest "wherein said API is a DLL file" as recited in claim 19. The Examiner states "APA does not explicitly teach the API is a DLL file. It would have been obvious to one of ordinary skill in the art to implement the API as a DLL because the DLL could be used in any number of systems." Paper No. 5, page 5. The Examiner admits that the references, APA, Russell and Dorn, do not teach an API that is a DLL file. Yet the Examiner asserts that by simply combining APA, Russell and Dorn, the limitation is taught. Applicants respectfully assert that the Examiner must first submit evidence

that this limitation is taught or suggested in any one of the above-identified references or is well known in the art. M.P.E.P. §2143-2144. Further, the Examiner must submit **objective evidence** and not rely on his own subjective opinion in support of combining APA, Russell and Dorn. *In re Lee*, 61 U.S.P.Q.2d 1430, 1434 (Fed. Cir. 2002). Accordingly, the Examiner has not presented a prima facie case of obviousness for rejecting claim 19. M.P.E.P. §2143.

C. Conclusion Regarding 35 U.S.C. § 103 Rejections.

As a result of the foregoing, Applicants respectfully assert that since there are numerous claim limitations not taught or suggested in the cited prior art, the Examiner has not presented a *prima facie* case of obviousness for rejecting claims 1-5, 7, 10, 15, 19-21, 23, 26 and 31 in view of the cited prior art.

It is noted that words are italicized only for emphasis. Words that are italicized are not meant to imply that only those limitations are not taught or suggested in the cited prior art.

II. ALLOWABLE SUBJECT MATTER:

Claims 6, 8, 9, 11-14, 16-18, 22, 24, 25, 27-30 and 32-34 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Paper No. 5, page 2. Applicants appreciate the indication of allowability of these claims.

III. DRAWINGS:

The Examiner notes that the PTO draftsperson has indicated informalities on the drawings that need to be corrected. Paper No. 5, page 2. Applicants will correct the informalities upon submission of formal drawings of Figures 1-4 after allowance.

IV. CONCLUSION

As a result of the foregoing, it is asserted by Applicants that claims 1-34 in the Application are in condition for allowance, and Applicants respectfully request an early allowance of such claims. Applicants respectfully request that the Examiner call Applicants' attorney at the below listed number if the Examiner believes that such a discussion would be helpful in resolving any remaining issues.

Respectfully submitted,

WINSTEAD SECHREST & MINICK P.C.

Attorneys for Applicants

By: _____

Robert Voigt
Reg. No. 47,159
Kelly K. Kordzik
Reg. No. 36,571

5400 Renaissance Tower
1201 Elm Street
Dallas, Texas 75270-2199
(512)370-2832

AUSTIN_1\214265\1
7036-P139US